HTWISH: RAPID HYBRID TESTING

- GUI based program originally written by Igor designed to test and analyze SVX3d hybrids

- Very convenient and fast: can create a database file with a large range of tests in less than 10 minutes

- My job has been to upgrade Htwish for SVX4 testing; it will be very useful in the coming months
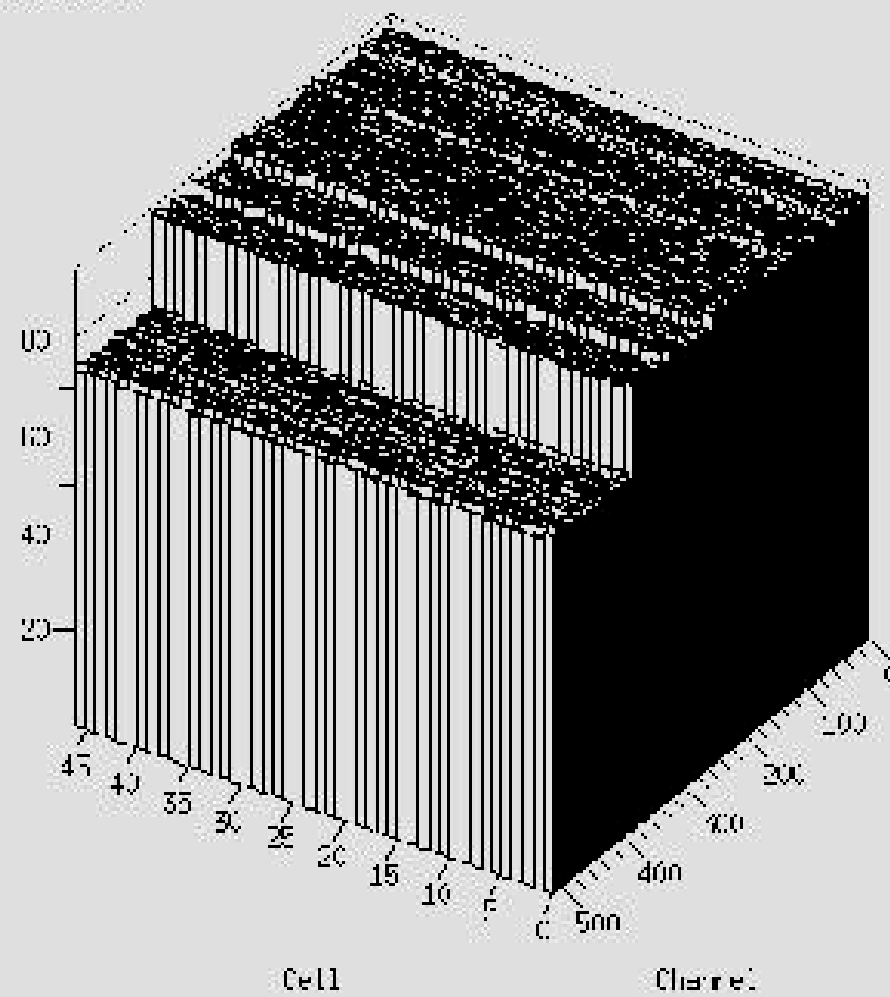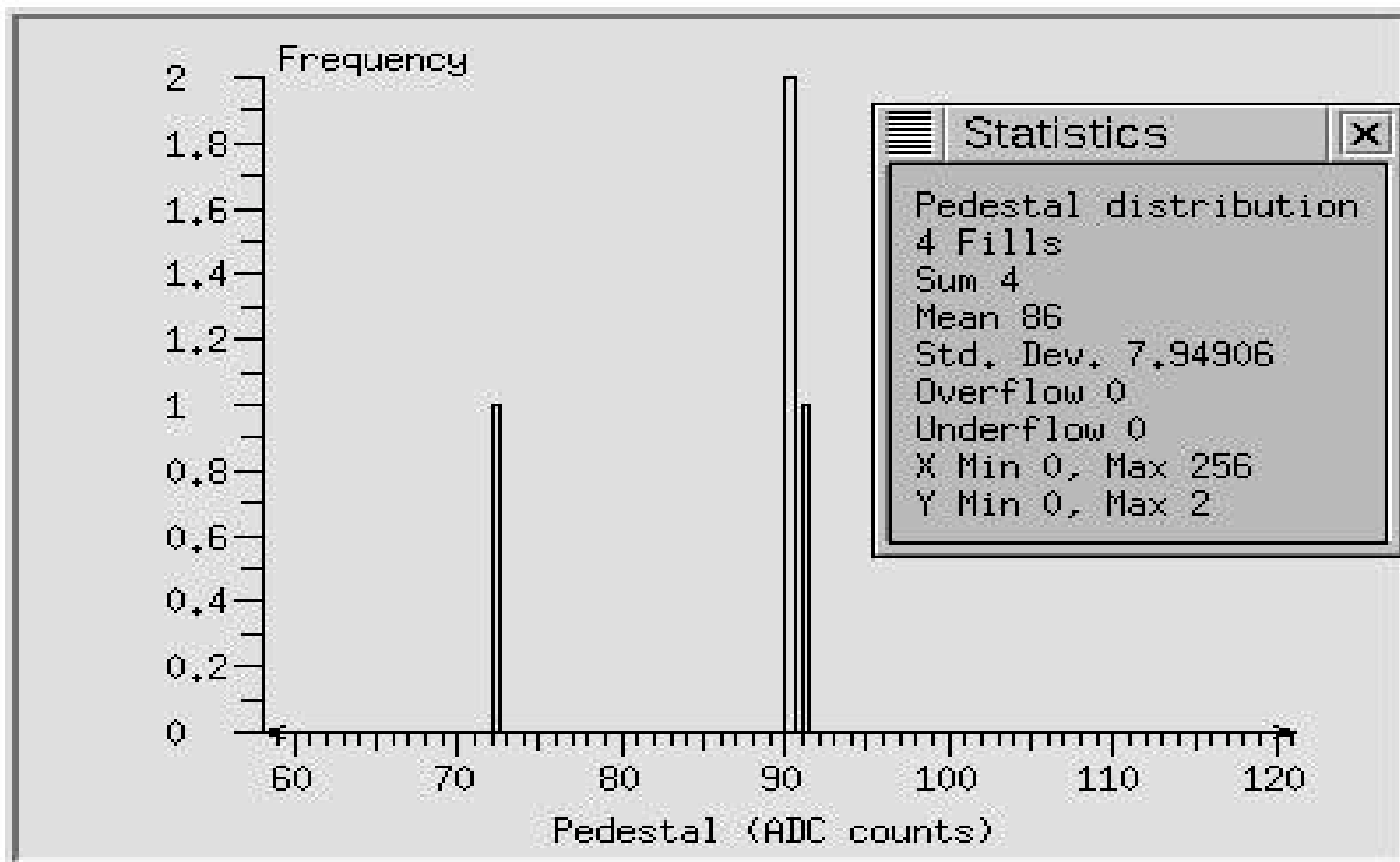
Outline of Discussion

- Introduction to the tests / current analysis results

- Discussion of how Htwish uses these results to automatically evaluate a hybrid

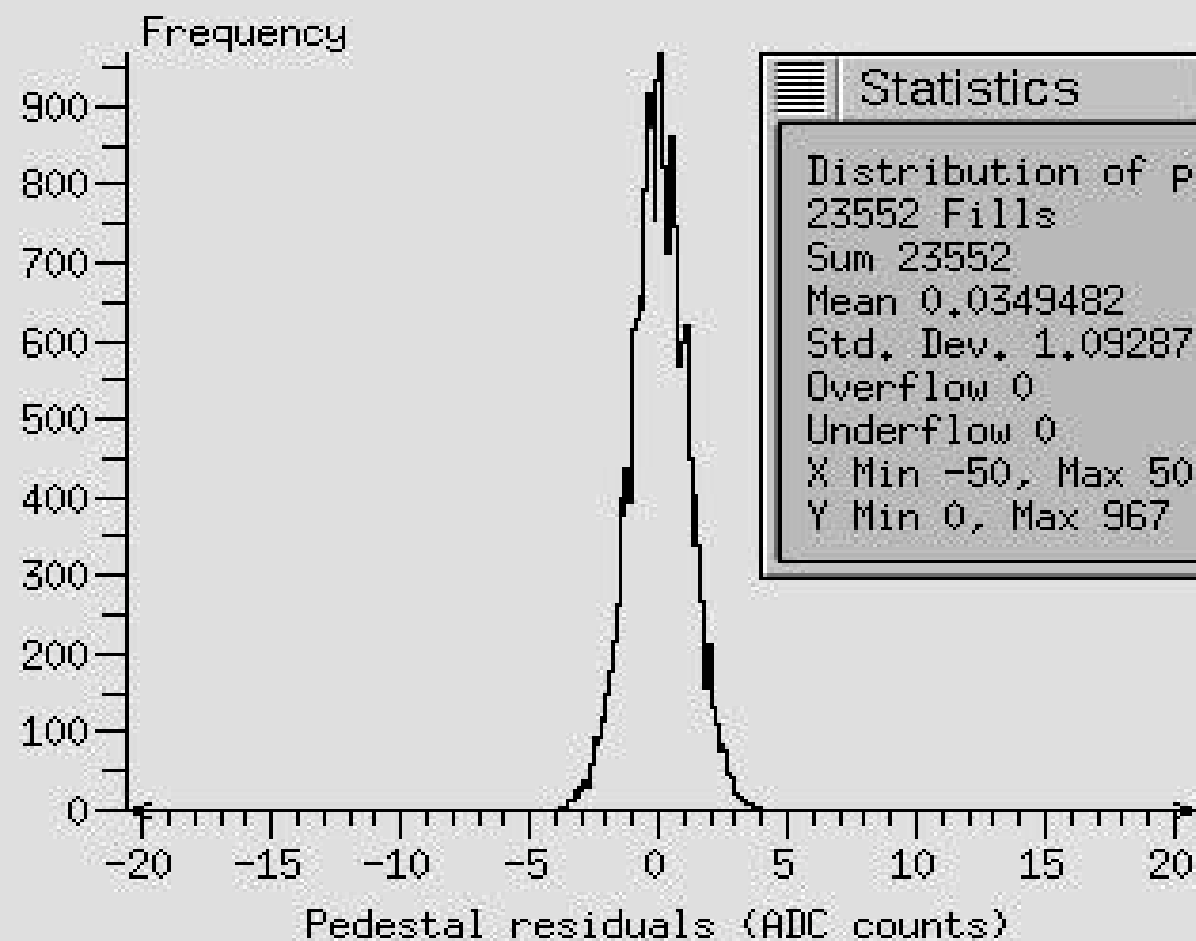- Where we are now with the program, and where we're going

Pipeline Pedestal Map

- Very fundamental; looks at the all the pedestal levels of pipeline capacitors on a hybrid (i.e. all channels and cells) while the chip is not digitizing/reading out

- Produces many raw data histograms which can be rapidly analyzed

- Results are used in a number of different ways for hybrid evaluation (more on this later…)

Frequency

Statistics ☒

Pedestal distribution
4 Fills
Sum 4
Mean 86
Std. Dev. 7.94906
Overflow 0
Underflow 0
X Min 0, Max 256
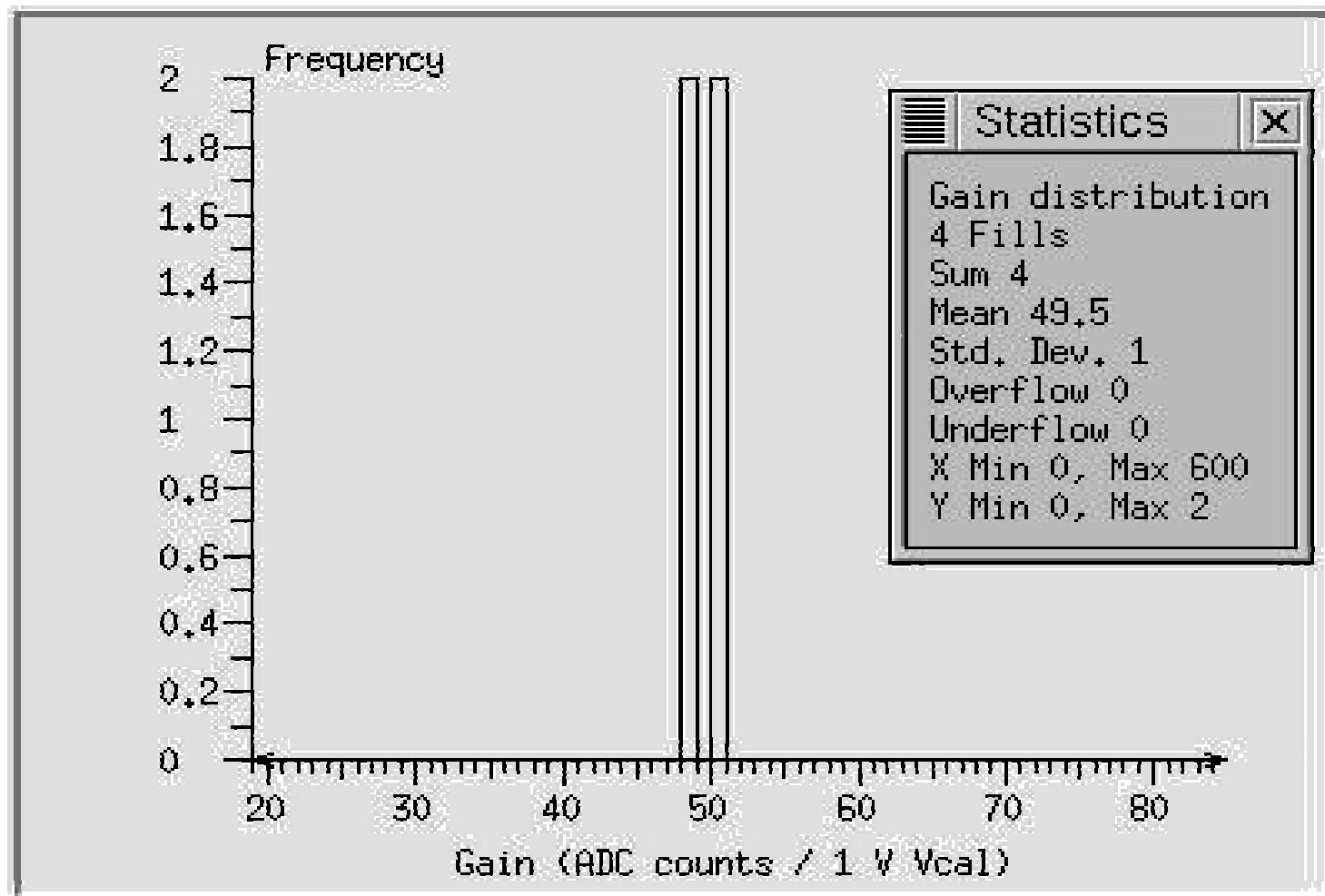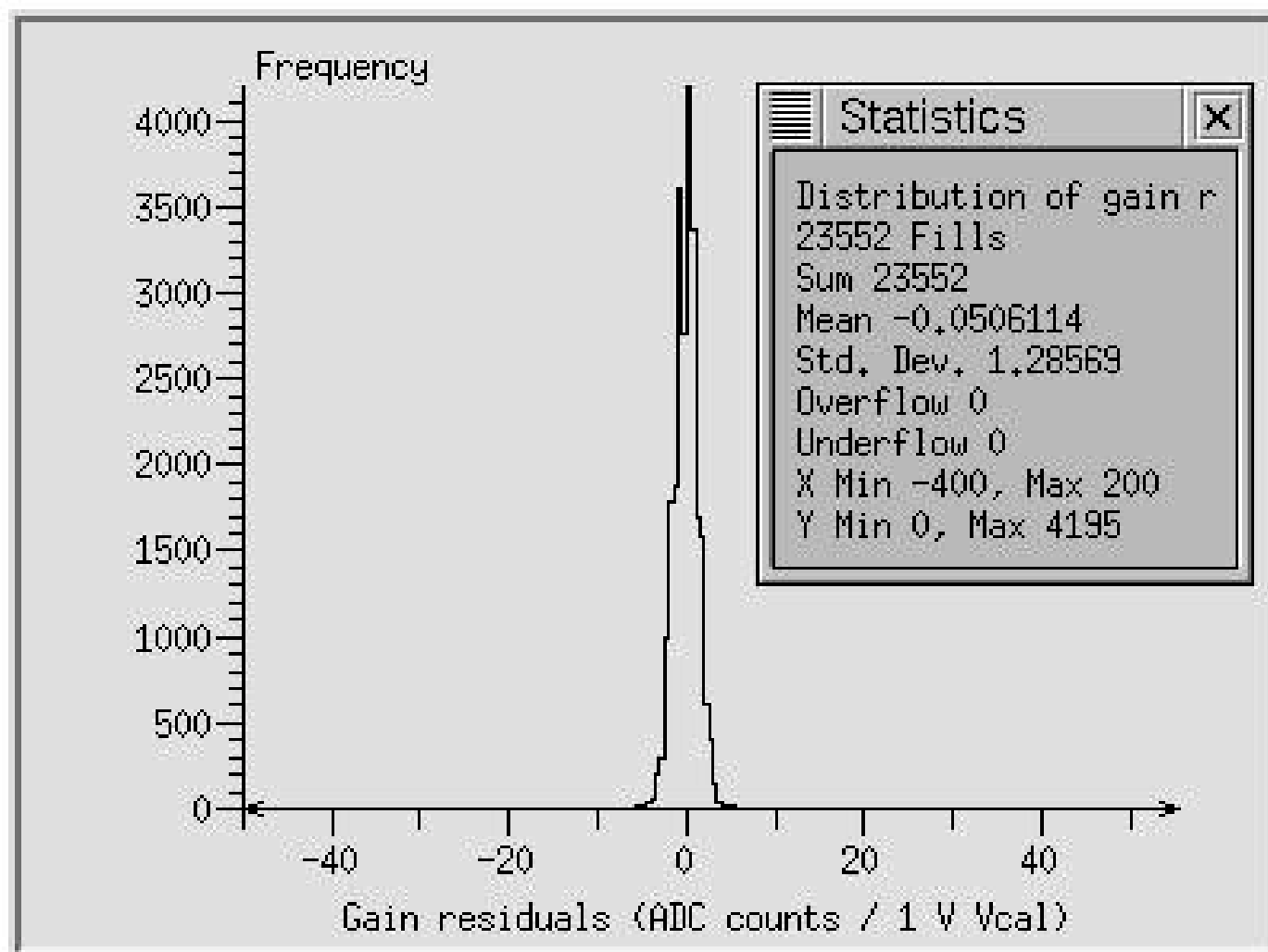Y Min 0, Max 2

Pedestal (ADC counts)

Frequency

Statistics

Distribution of ped:
23552 Fills
Sum 23552
Mean 0.0349482
Std. Dev. 1.09287
Overflow 0
Underflow 0
X Min −50, Max 50
Y Min 0, Max 967

Pedestal residuals (ADC counts)

Pipeline Gain Map

- Injects into all channels at both 0.5 and 0.7 volts

- Calculates gain in [ADC counts/V] for all pipeline capacitors

- Current drawback: calculates gain based on the nominal VCAL voltage, not what the chip actually "sees"

Frequency

Gain (ADC counts / 1 V Vcal)

Statistics

Gain distribution
4 Fills
Sum 4
Mean 49.5
Std. Dev. 1
Overflow 0
Underflow 0
X Min 0, Max 600
Y Min 0, Max 2

Frequency

Statistics

Distribution of gain r
23552 Fills
Sum 23552
Mean -0.0506114
Std. Dev. 1.28569
Overflow 0
Underflow 0
X Min -400, Max 200
Y Min 0, Max 4195

Gain residuals (ADC counts / 1 V Vcal)

Pipeline Delay

- With the L1A trigger in a fixed bucket, steps the trigger latency from 1-42, and reads out at every latency setting

- Designed to make sure that cell IDs read out are what we expect

- No histograms in case of success; if failure, a standard "Error Table Histogram" is produced

Risetime

- Used to examine the behavior of the preamp at different bandwidth settings
- A little less useful for the SVX4 than the SVX3, due to the SVX4's greater preamp speed
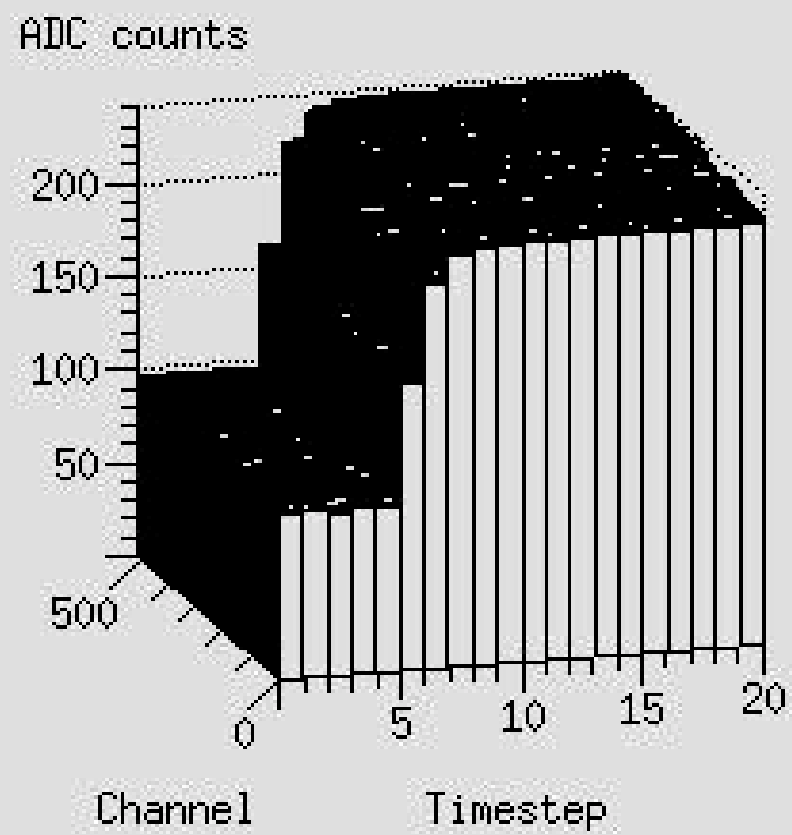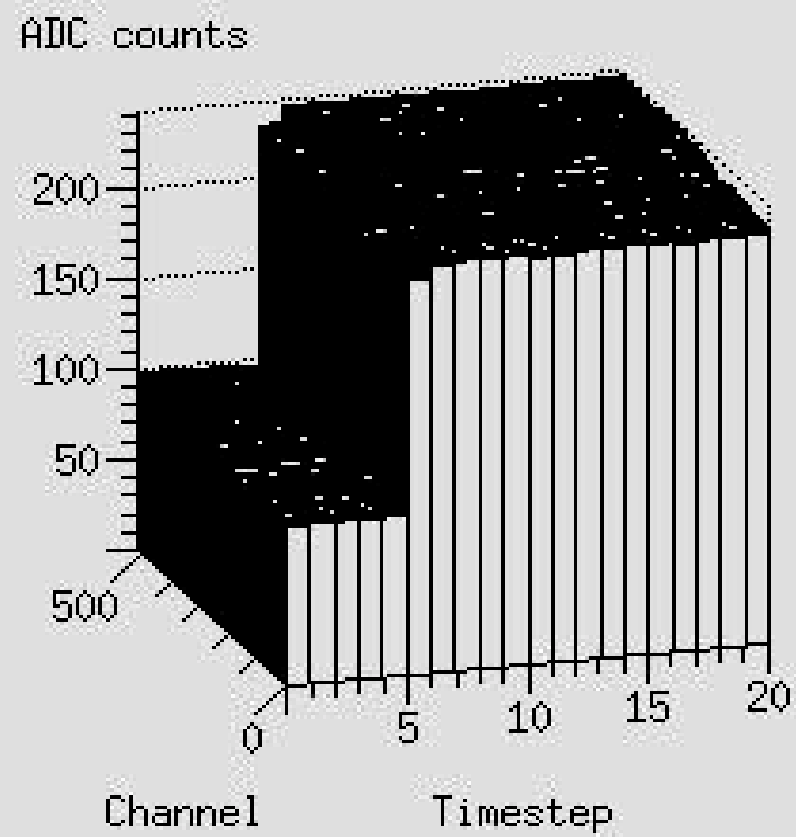
ADC Limits

- A test designed to insure that no bits are stuck in the ADC latch
- Digitize for different settings of counter modulo
- Make sure pedestal level is always exceeding counter modulo
- Methods by which this is achieved are different for our SVX4 version of Htwish

Dynamic Common Mode Suppression

- Idea is to see how many channels on a chip need to be injected before chip is fooled into subtracting off the injected, rather than the pedestal, channels

- Test appears promising, but technical problems with reading out a heavily injected chip have been occurring
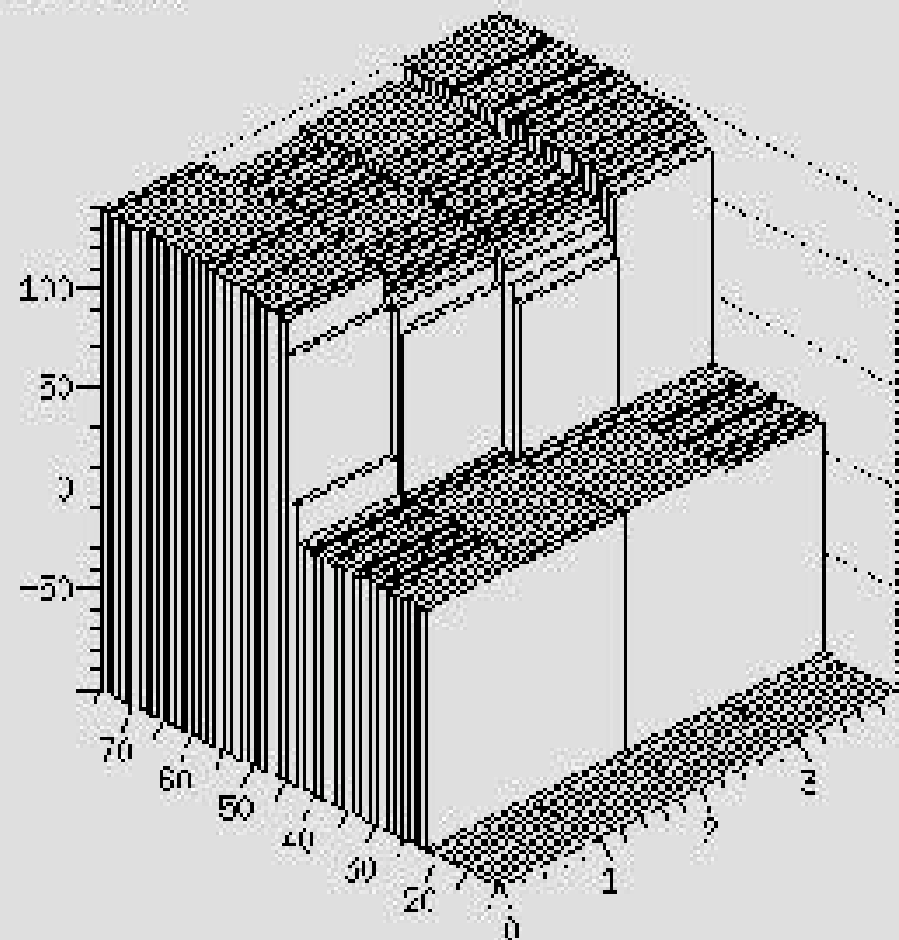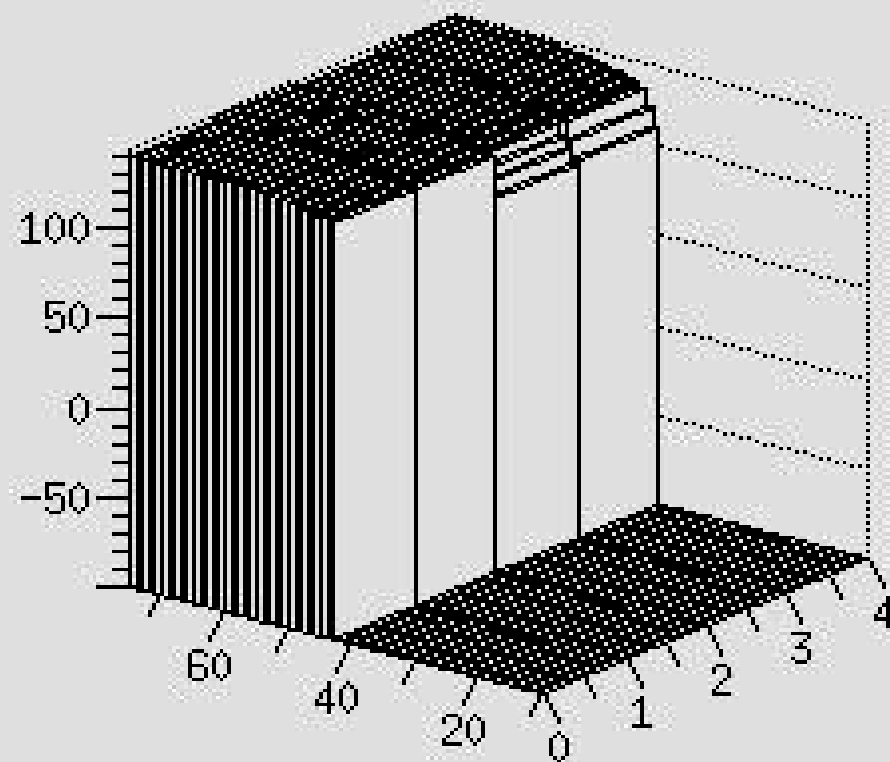
HDC counts

Channels at pedestal   Chip number

Sparsification

- Check to make sure the chip can distinguish between pedestal and injected channels by removing "read-all" mode
- Test performed both with and without read-neighbor feature
- Attenuated version of events read out poses some (minor) coding challenges

Additionally…

- **Multihit test**: idea is to check for proper cell IDs in case of four closely spaced L1A triggers occur

- **Serial line test**: check to see if initialization bits are correctly read out on the priority-out line

Getmap

- Much like the standalone program Marc and I have been using, but much quicker!

- Examines deadtimeless operation by stepping through the buckets during a readout sequence. Three flavors: regular, sparse mode, and Dynamic Common Mode Suppression

- Creation of test data works fine, but analysis code needs to be improved

Htwish Analysis

- Htwish will apply the results of these tests to evaluate the hybrid
- Evaluation is based on cuts, which are currently undergoing an evolution to reflect the behavior of SVX4 hybrids
- Analysis is hierarchical, and presented as a "Quality Tree"
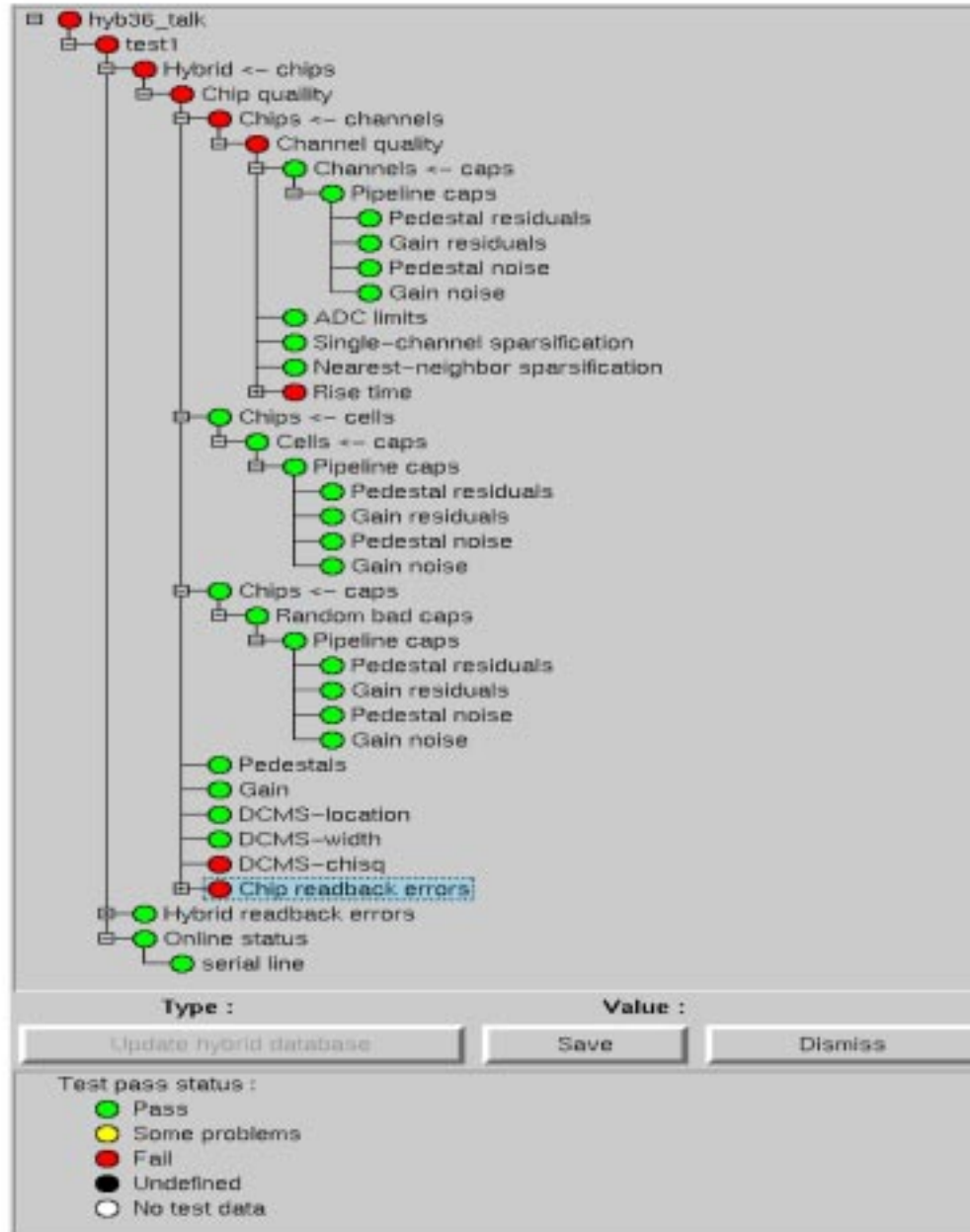
Currently

- Htwish can properly produce test data. ASCII screen dumps indicate that this data is what we expect.

- Much analysis code can be improved: data integrity checks need to become more "intelligent", and cuts will also be improved as we learn more about our hybrids

- Nonetheless, it has already proved useful in evaluating LBL hybrids

Future

- Near term: problems just mentioned will be solved in the next week or so. Some of the testing may be streamlined

- Htwish will be very useful for rapid testing of the new SVX4 version

- Likewise in the long run, will be indispensable for testing ~1500 hybrids